Automation of Indian Postal Documents written in Bangla and English

Szilárd Vajda, Kaushik Roy, Umapada Pal, Bidyut B Chaudhuri, Abdel Belaïd

Automation of Indian Postal Documents written in Bangla and English

S. Vajda[1], K. Roy[2], U. Pal[2], B. B. Chaudhuri[2] and A. Belaid[1]
[1]*LORIA Research Center, Campus Scientifique, 615, rue du jardin botanique, B.P. 239, 54506 Villers-Lès-Nancy, France*
[2]*Computer Vision and Pattern Recognition Unit, Indian Statistical Institute, Kolkata-108, India,*
Email: bbc@isical.ac.in, abelaid@loria.fr

Abstract

In this paper, we present a system towards Indian postal automation based on pin-code and city name recognition. Here, at first, using Run Length Smoothing Approach (RLSA), non-text blocks (postal stamp, postal seal, etc.) are detected and using positional information Destination Address Block (DAB) is identified from postal documents. Next, lines and words of the DAB are segmented. In India, the address part of a postal document may be written by combination of two scripts: Latin (English) and a local (State/region) script. It is very difficult to identify the script by which pin-code part is written. To overcome this problem on pin-code part, we have used two-stage artificial neural network based general scheme to recognize pin-code numbers written in any of the two scripts. To identify the script by which a word/city name is written, we propose a water reservoir concept based feature. For recognition of city names, we propose an NSHP-HMM (Non-Symmetric Half Plane-Hidden Markov Model) based technique. At present, the accuracy of the proposed digit numeral recognition module is 93.14% while that of city name recognition scheme is 86.44%.

1. INTRODUCTION

Postal automation is a topic of research interest for last two decades and many pieces of published article are available towards automation of non-Indian language documents [18]. Many systems are also available for address recognition in several countries like USA, UK, Japan, Germany etc. However, no system is available for address recognition of Indian postal documents. Because of the multi-lingual and multi-script behaviour, system development towards postal automation for a country like India is more difficult than such task in other countries. In India, there are about 19 official languages and an Indian postal document may be written in any of these official languages. Moreover, some people write the destination address part of a postal document in two or more language scripts. For example, see Fig. 1, where the destination address is written partly in Bangla script and partly in English. In addition, we have seen that in 36.2% of the documents, pin-code is either absent or partially written. Bangla is the second most popular language in India and the fifth most popular language in the world. About 220 million people in the eastern part of Indian subcontinent speak in this language. Bangla script alphabet is used in texts of Bangla, Assamese and Manipuri languages. In addition, Bangla is the national language of Bangladesh.

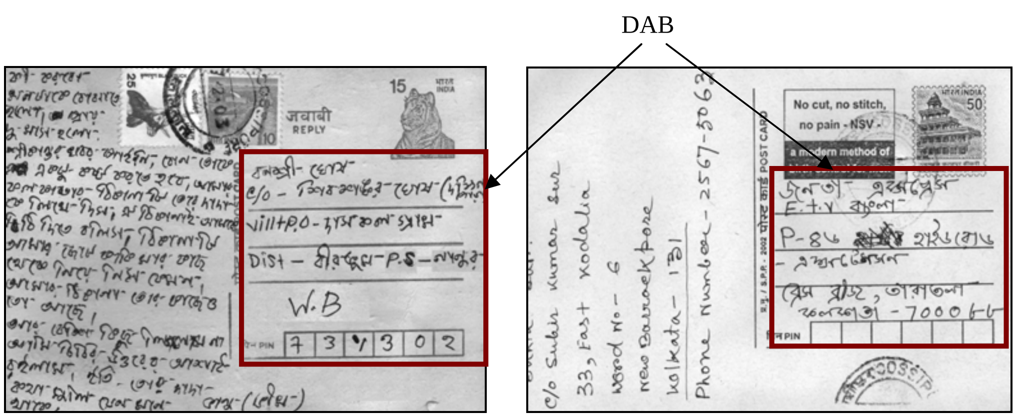


Fig. 1. Examples of Indian postal document.

Indian postal code (pin-code) is a six-digit number. Based on this six-digit pin-code (zip-code) configuration we cannot locate a particular village in a post office contrary to western countries where the pin-code gives a precise location of the address containing even the street and number. We can only locate a post office with this pin-code. There are eight postal zones in the country. The first digit of pin-code indicates one of these regions. The first two digits together indicate the sub-region or one of the postal circles. The first three digits together indicate a sorting revenue district. The last three digits refer to the delivery Post Office. Representation of first two digits of pin-code is shown in Table 1.

Table 1: Representation of first and second digit of Indian pin-code.

The first digit and covering region		First two digit and their representation	
First Digit	Region	First 2 Digit	States/Circle Covered
1 2	Northern	11	Delhi
		12 to 13	Haryana
		14 to 16	Punjab
		17	Himachal Pradesh
		18 to 19	Jammu & Kashmir
		20 to 26	Uttar Pradesh
		27 to 28	Uttaranchal
3 4	Western	30 to 34	Rajasthan
		36 to 39	Gujarat
		40 to 44	Maharashtra
		45 to 48	Madhya Pradesh
		49	Chattisgarh
5 6	Southern	50 to 53	Andhra Pradesh
		56 to 59	Karnataka
		60 to 64	Tamil Nadu
		67 to 69	Kerala
7 8	Eastern	70 to 74	West Bengal
		75 to 77	Orissa
		78	Assam
		79	North Eastern
		80 to 85	Bihar
		86 to 88	Jharkand

In India, there is a wide variation in the types of postal documents. Some of these are post-card, inland letter, aerogram, special envelope, ordinary envelope, etc. Post-card, inland letter, special envelopes are sold from Indian post offices and there is a pin-code box of six-digit to write pin number in the postal document obtained from post office. Like some other countries here also, because of educational backgrounds there is wide variation in writing style and medium. In some documents, we may find partial pin-code instead of full pin-code. For example, Kolkata-32 (or Kol-32) is written instead of Kolkata – 700032. In addition, some people sometimes do not mention pin-code on Indian postal documents. Thus, development of Indian postal address reading system is a challenging problem.

In this paper, we describe a system towards Indian postal automation where at first, using Run Length Smoothing Approach (RLSA) [24] and characteristics of different image components, the postal stamp/seal parts are detected and removed from the documents. Next, based on positional information, the DAB region is located. Pin-code written within the pin-code box is then extracted from the DAB region. Using a two-stage neural network [19], the Bangla and English numerals of the pin-code part are recognized. For documents where pin-code is either absent or partially written we need to recognize the city name or post office name as well as the nearest town name. For this purpose, we first segment DAB into text lines and words. Next, using water reservoir concept [15] based feature, word-wise script identification is done. After a differential height normalization of word images, a 2D stochastic approach (NSHP-HMM) [6, 22] is employed to recognize script-wise words directly on image pixels. To increase the system accuracy, structural features are embedded with the pixel column information considered by the NSHP-HMM word recognition process. A part of this paper has been published in IWHFR-2004 [21] but script identification, city-name recognition methods were not discussed there.

2. PREPROCESSING

2.1 Data collection and noise removal

Document digitization for the present work has been done from real life data collected from Cossipore Post Office of North Kolkata circle, West Bengal, India. We used a flatbed scanner (manufactured by UMAX, Model AstraSlim) for digitization. We have collected 7500 data from post office for our experiment of the proposed work. The images are in gray tone with 300 dpi and stored as Tagged Information File (TIF) Format. We have used a two-stage approach to convert them into two-tone (0 and 1) images. At the first stage, a local thresholding algorithm due to Palumbo et. al. [16], is applied to the images. The output binary image is stored on a binary array without any change to the original array, as we need them for final binarization. Due to the window size, any solid or fully filled part of the document greater than window size (W) may become hollow or broken by this method. To overcome the difficulty in local thresholding we have used Run Length Smoothing Approach (RLSA) [24] on this pre-binarized image to get the homogeneous area as single component. After applying component labelling of smoothed image, we map each component on the original image and the final binarized image is obtained using a histogram based global binarizing algorithm on the components [4] (Here ‘1’ represents object pixel and ‘0’ represents background pixel). The digitized document images may be skewed and we used Hough transform based method due to Shi and Govindaraju to de-skew the documents [23]. The digitized image may contain spurious noise pixels and irregularities on the boundary of the characters, leading to undesired effects on the system. We have used a smoothing technique due to Chaudhuri and Pal [4] to correct the noise.

2.2 Statistical analysis

We calculated some simple statistics on the collected postal data to get an idea of different components useful for Indian postal automation. Some of the computed statistics are: percentage of postal documents having pin-code box, presence of the pin-code written in the pin-code box, percentage of writers who write all the digits of pin-code, postal document without pin-code on DAB. Also, we analyzed distribution of different types of postal documents like percentage of envelope, inland letters, postcards, etc. Percentage of postal documents where address part is handwritten, document written in two or more scripts and position of the address part is also analyzed. The primary statistics we obtained from 7500 data are as follows:

1. 65.69% of the postal documents contain pin-code box.
2. 73.59% people write pin-code within the pin-code box.
3. 63.8% people write all the digits of the pin-code (irrespective of pin-code box).
4. 13.49% writers do not mention pin-code on postal documents.
5. 10.02% touching characters are present in the pin-code number.
6. 05.83% document addresses are printed and the rest are handwritten.
7. 24.62% of the addresses are written in Bangla, 65.37% in English and 22.04% address are written in two language scripts (English and local state language).
8. 87.6% cases the address portion is written at the bottommost, 72.3% at the rightmost and 70.6% at right bottommost position.
9. Among the collected postal documents 13.41% are envelopes, 31.09% postcards and 15.76% Inland letters (a kind of letter that can be sent anywhere within India).

2.3 Postal stamp detection and deletion

The next step consists of stamp and graphics extraction from the postal documents. There are many techniques for text/graphics separation. Here we used a combined technique as follows. At first, simple horizontal and vertical smoothing operations of RLSA are performed [24] on the image. The two smoothing results are then combined by logical AND operation. The results after horizontal, vertical and logical AND operation of Fig. 2(a) are shown in Fig. 2(b), (c) and (d), respectively. The result of logical AND operation is further smoothed to delete the *stray* parts (see Fig.2(e)). On this smoothed image, we apply component labelling to get individual blocks and the blocks having bigger size in height as well as width are considered for graphics block. Because of this smoothing sometimes a text line may be included in a graphics block. For example, see Fig. 2(f) where such a block obtained from Fig. 2(e) is shown. To extract the text line from the graphics block, we analyze the shape of the block. When we write a text line generally we provide some blank space between two consecutive text lines. As a result when a text line touches a graphics block we can find a white patch of blank space in the block [as it can be seen in the component shown in Fig.2(f)]. We find this white patch information to detect whether a text line touched a graphics block or not. For this purpose, we compute histogram of the black pixels row-wise and based on peak-valley position of the histogram, we segment the block to get touching text line. See Fig. 2(g) where segmentation result of a touching block is shown.

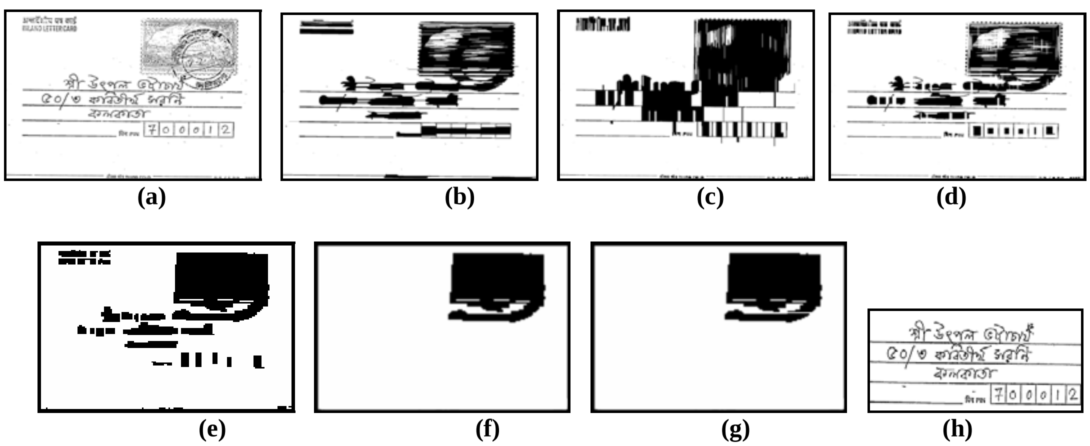


Fig. 2. (a) An example of postal document image obtained from an Inland letter. (b) Horizontal run-length smoothing of Fig. (a). (c) Vertical run-length smoothing of Fig. (a). (d) Logical AND of (b) and (c). (e) Smoothed version Fig. (d). (f) Detected irregular block. (g) Segmentation results of irregular block. (h) Detected DAB part of Fig.2(a).

Finally, for each segmented block, we find its boundary and check the density of black pixels over the corresponding boundary area on the original image. We note that for postal stamp/seal block the densities of black pixels are very high compared to that of text line block. In addition, we noticed that the postal stamp/seal block contains many small components, whereas such small components are not present in other blocks. Based on the above criteria, non-text block are detected. After detection of a postal stamp/seal block, we delete that block from the documents for future processing.

2.4 DAB detection

As mentioned in the Section 2.2 the address on the postal document is generally written in the manner that DAB will be in the right lower part of the documents. For detection of DAB from the text part, we divide the text part into blocks by iteratively partitioning it in horizontal and vertical direction. Before horizontal (vertical) partitioning we smooth the partition vertically (horizontal) so that the text block itself does not get split. This process is repeated until the text part cannot be split. Next, each block is analyzed and the right bottommost block, is considered as DAB and extracted from the postal document for further analysis. Extracted DAB of Fig. 2(a) is shown in Fig. 2(h).

2.5 Script identification

Because of multi-lingual and multi-script behaviour, a postal document may be written by more than one script. There are only a few works reported on script separation of handwritten documents in Indian script. Recently Zhou et al. [29] proposed a scheme of identifying script of a bi-lingual postal document written in Bangla-English and they worked on document level, which does not work for a mixed postal document as found here. To recognize a word correctly, it is necessary to feed it to the OCR of that language in which the word image belongs. Therefore, we have to identify the script of each word for recognition. Here we use the method due to Pal and Datta [14] for text line and word segmentation. Before going to describe different features for script identification, here we discuss water reservoir principle on which some of the features for script identification are based.

2.5.1 Water reservoir principle: The principle of water reservoir property is as follows. If water is poured from a side of a component, the cavity regions of the component where water will be stored are considered as reservoirs [15]. While writing by hand, characters in a word touch one another and create a large space (cavity). This space generates reservoir. By top (bottom) reservoir of a component, we mean the reservoirs obtained when water is poured from top (bottom) of the component. A bottom reservoir of a component is visualized as a top reservoir when water is poured from top after rotating the component by 180°. Examples of top and bottom reservoir are given in Fig. 4(c).

2.5.2 Joining of isolated characters and computation of busy-zone: Script identification scheme mainly depends on water reservoir concept based features, and water reservoir is obtained when neighbouring characters in a word are touching. However, some people write a word in isolated fashion where characters do not touch each other. As a result, water reservoirs are not generated and hence the above scheme will not work properly. To take care of this situation, we join isolated characters in a word. Joining procedure of the characters in a word is as follows. Here we first take one component of the word image and grow it around its boundaries until it touches its nearest neighbour component. This touching point is noted for joining. Let this point be X_1. Considering this point as centre, we draw a circle of radius equal to number of times the component was grown to touch the neighbour component. The point where the circle touches the first component is noted. Let this point be X_2. Two points X_1 and X_2 are joined with width equal to the stroke width to get touching component. For illustration, see Fig. 3. The above process is repeated until all the components of a word are connected. An example of isolated characters of a Bangla word and its joined version is shown in Fig. 4a-b).

(a) (b)

Fig. 3. Joining of isolated characters is shown. (a) A word containing two isolated characters and the grown version of the left component (in gray). (b) Word after joining.

Busy-zone of a word is the region of the word where maximum parts of its characters lie. So, maximum information regarding the script can be extracted from the busy zone. Most of our features used for script separation are dependent on busy zone and we detect the busy-zone by using water reservoir principle. Here, at first all top and bottom reservoirs are detected. We calculate the average height of all top and bottom reservoirs. Top and bottom reservoirs, whose height is less than 1.25 times the average reservoirs height, are filled by black pixels. This threshold value is calculated from experiments. In addition, all the loops are filled with black pixel before computing the busy-zone. Filled-up version of the image of Fig. 4(a) is shown in Fig. 4(c). After filling the reservoirs and loops with black pixels, we compute the busy-zone height of this filled-up image as follows. At first, we compute horizontal projection profile on this filled-up image. We draw a vertical line at the middle of the horizontal projection profile. Let this line be XY as shown in Fig. 4(d). The portion of the XY that belongs to projection profile is marked by p and q. The distance between p and q is the busy-zone height of a word image. The region within the horizontal lines passing through p and q gives us the busy-zone. Busy-zone of the word of Fig. 4(a) is shown in Fig. 4(e).

Identification of Bangla and English scripts is done mainly using water reservoir principle along with Matra/Shirorekha based feature, and position of small components etc. These features are described below.

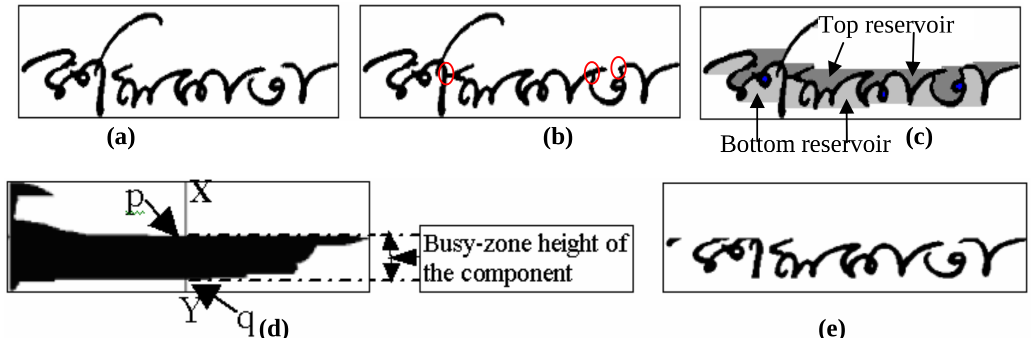


Fig. 4. (a) Original Image, (b) Image after joining, joining portion is marked by small circles (c) Image after filling of top, bottom reservoir and loop, (d) Computation of busy-zone, (e) Busy-zone area of the image shown in (a).

2.5.3 Matra/Shirorekha based feature: The longest horizontal run of black pixels on the rows of a Bangla text word will be much longer than that of English script. This is so because the characters in a Bangla word are generally connected by Matra/Shirorekha (see Fig. 5). Here row-wise histogram of the longest horizontal run is shown in the right part of the words. This information has been used to separate English from Bangla script. Matra feature is considered to be present in a word, if the length of the longest horizontal run of the word satisfies the following two conditions: (a) if it is greater than 45% of the width of a word, and (b) if it is greater than thrice of the height of busy-zone. These two conditions are obtained from examining the characteristics of Bangla and English scripts.

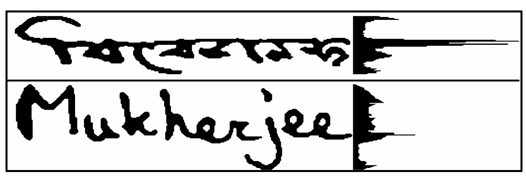


Fig. 5. Example of longest horizontal run to decide Matra/Shirorekha features.

2.5.4 Water reservoir based feature: In English text it may be noted that there are many big top reservoirs [15]. On the other hand, Bangla text has many big bottom reservoirs. For example, see Fig. 6. The ratio (r) of the area of the top reservoirs to that of bottom reservoirs of a word image is used as feature (by area of a reservoir we mean number of white pixels inside the reservoir). Here we note that the value of r is greater than one for English and less than one for Bangla script. Another distinct feature between Bangla and English is that the base line (the line passing through the average of all the base points (the deepest point) of the reservoirs) lies in the lower half of the busy-zone for English, whereas it is in the upper half for Bangla. For illustration, see Fig. 6.

2.5.5 Feature based on the position of small component: Here we take all the components whose height and width is less then twice of the stroke width (R_w) of the image and compare their position with respect to the busy-zone. If such components lie completely above or below the busy-zone, then the number and position of such components are used as a feature. This feature is selected because in English we find some characters with disjoint upper part (like dots of i and j) and in Bangla we find some characters with disjoint lower part (like dots of র,ড়, etc.). Here R_w is the stroke width of a component. In other words, R_w is the statistical mode of the black run lengths of the components. The value of R_w is calculated as follows. The component is scanned both horizontally and vertically. Let from this component we get n different run-lengths r_1, r_2,..r_n with frequencies f_1, f_2 ...f_n, respectively. In this case, the value of $R_w = r_i$ where $f_i = \max(f_j)$, j = 1...n.

Based on the above features, we use a tree-classifier for script identification. The first feature used in the tree, is Matra/Shirorekha based feature, because it is noted that the probability of occurrence of Matra/Shirorekha in a Bangla handwritten word is about 67%, and in printed word it is 99.97%. So, the use of Matra/Shirorekha based feature at the top of the tree classifier is justified. If this feature exists, then the word is identified as Bangla script. Otherwise, the word is checked for the water reservoir based feature. If the base line of most of the bottom reservoir lies in upper part of the busy-zone, then the word is treated as Bangla script. Else, the word is checked by ratio of reservoir area. If the ratio of the areas of top and bottom reservoirs is greater than 1.25, then the word is identified as English and if it is less than 0.75 then the word is identified as Bangla. Lastly, the word is tested by the position of small isolated component. If there exist only small components, above (below) the busy-zone, it is identified as English (Bangla). Else, the word is left as confused and rejected by the system.

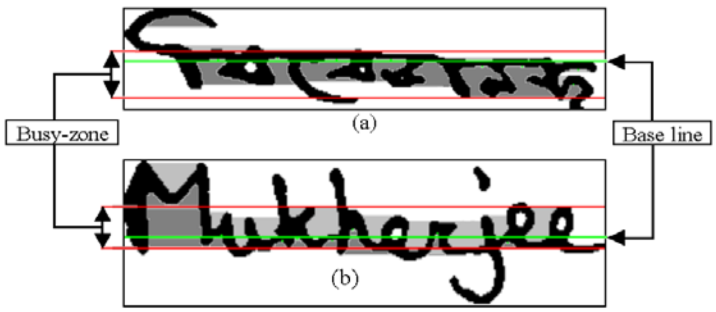


Fig. 6. Water reservoirs are shown on (a) Bangla and (b) English words. Top and Bottom reservoirs are marked by different gray shades.

2.6 Pin-code box detection and extraction.

People generally write the destination pin-code inside the pin-code boxes. Here, at first, we detect whether there is a pin-code box or not. If it exists, our method will extract the pin-code from the box if pin-code is written within it.

For pin-code box extraction, we apply component labelling and select those components as candidates, which satisfy the following criteria. A component is selected as candidate component if (i) the length (L) of the component is greater than five times the width (W) of the component and (ii) the length of the component is less than seven times the width of the component. Length and width of a pin-code is shown in Fig.7(a). Since an Indian pin-code box contains six square boxes, the length of a box component will be about six times the width of the component. Based on this principle we choose the candidate component. Let X be the set of these selected components. From the candidate components, we decide the pin-code component as follows. We scan each column of a selected component from top and as soon as we get a black pixel, we stop and note the row value of this point. Let t_i be the row value of the i^{th} column obtained during top scanning. Similarly, we scan each column of the selected component from bottom and as soon as we get a black pixel, we stop and note the row value of this point. Let b_i be the row value of the i^{th} column obtained during scanning from bottom. We compute the absolute value of $b_i - t_i$, for all columns. The selected component with width W satisfying $|(b_i - t_i) - 2R_w| \leq W \leq |(b_i - t_i) + 2R_w|$, for all i=1 to L, is chosen as pin-code box component (R_w is the stroke width of a component). If no such candidate component is obtained, then we assume that there is no pin-code box.

After pin-code box detection, we check whether a numeral written in the text box crossed the pin-code box or not. To check it we compute the two horizontal lines of the pin-code box as well as the bounding box of the pin-code. Two horizontal lines and the bounding box, of a pin-code box shown in Fig. 7(a), are shown in Fig. 7(b). If the upper and the lower boundary of the pin-code box coincides with the two horizontal lines of this pin-code box then we assume that any of the numerals written in the pin-code box does not cross the pin-code box. Otherwise, they cross the pin-code box. If there is no crossing, we detect the vertical line from the pin-code box and delete both vertical and horizontal lines from the pin-code box. Next, depending on the positions of the vertical lines, the pin-code numerals are extracted from left to right to preserve the numerals order occurrence. Pin-code box extracted from Fig. 2(f) by the proposed algorithm is shown in Fig. 7(a). In addition, pin-code numerals extracted from Fig. 7(a) are shown in Fig. 7(b).

Fig. 7. (a) Extracted part of pin-code box from the DAB shown in Fig. 2. (f) Extracted pin-code numerals from the pin-code box.

When there is a crossing, we find the position where crossing is made and this position is decided using the positional information of the horizontal lines of the pin-code box and its bounding box. We examine the area between the two horizontal lines and the bounding box of the pin-code to find the portion where black pixels are present. See Fig. 8(c) where such a portion is shown by a small box. From the lowermost point of this portion we trace the boundary in clock-wise direction and find the position (say l_c) where it meets the lower horizontal line of the pin-code box. Again from the lowermost point of this portion we trace the boundary in anti-clock-wise direction and find the position (say l_a) where it meets the lower horizontal line of the pin-code box. During clock–wise (anti-clock-wise) tracing we also noted a contour point at a distance R_w from l_c (l_a) to get the orientation of the crossing part. Let p_c (p_a) is the corresponding point of l_c (l_a). We compute the slope m_c (m_a) of p_c and l_c (p_a and l_a). l_c , l_a, p_c, p_a, m_c and m_a are shown in an image in Fig. 8(d). We also compute the average slope of m_c and m_a. This average slope gives the orientation of the crossing part and based on this slope information we extract the crossing part from the pin-code box. Pin-code numerals extracted from Fig. 8(a) are shown in Fig. 8(e).

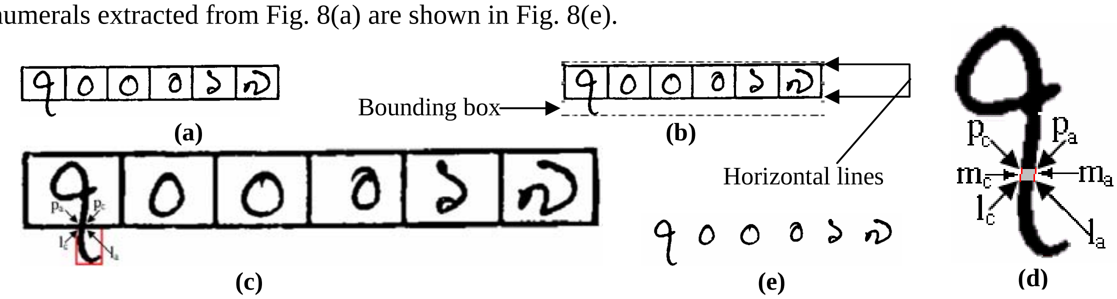


Fig. 8. Extraction of pin-code numeral from pin-code box having extended part. (a) Extracted pin-code. (b) Bounding box and horizontal line are shown. (c) The extended park (marked in gray) along with crossing points are shown. (d) Extracted part is shown.(e) Extracted numerals from Fig.8(a) are shown.

3. NUMERAL RECOGNITION

Research on recognition of unconstrained handwritten numerals has made impressive progress in Roman, Chinese, Japanese and Arabic scripts [17,28], and the researchers have proposed various approaches for this purpose. One of the widely used approaches is based on the Neural Network (NN) [21]. Statistical approach is also applied to numeral recognition [25]. Integration of structural and statistical approach has also been considered numeral recognition[3]. It is less sensitive to pattern noise and distortion but modelling of statistical information is a tedious task.

Many techniques have been proposed for Bangla handwritten numeral recognition [1,2,7,12,13]. Dutta and Chaudhuri [7] proposed a two stage feed-forward NN based scheme for the same. They used primitives and structural constraints between the primitives imposed by the junctions present in the characters for this purpose. Pal and Chaudhuri [12] proposed a structural feature based approach for Bangla handwritten numeral recognition. Bhattacharya et al. [2] used a modified Topology Adaptive Self-Organizing NN to extract a vector skeleton from a binary numeral image. Basu et al. [1] used a two-pass approach for recognition of handwritten numerals. Wen et. al. [27] proposed a system for postal automation of Bangladesh post based on handwritten Bangla numeral recognition.

Because of quality of postal documents and the writing media, numerals of the pin-code of Indian postal documents may be poor in quality. To handle such documents we have proposed an NN-based numeral recognition system based on the pixel information of the image. Sometimes because of poor document a numeral may be disconnected. Analyzing the morphological structure features of the numerals, broken parts of the numerals are connected to improve recognition performance [20].

To generate the structural features for connecting broken parts of the numerals, we use contour smoothing and linearization. For any component of the image, using Freeman chain code based edge-tracing algorithm the outer contour of component is extracted. The contour is then smoothed and converted to lines consisting of ordered pixels. Next, depending on the value of the direction codes of two consecutive lines the structural codes are assigned to the start or end points of the linearized lines of the contours. The structural points describe the convex or concave changes in different chain code direction along the contour. So, they can be used to represent the morphological structures of a contour. After detection of the structural points, the binary image is thinned to get the end and junction points. Next, based on some predefined criteria, we select a pair of end points for joining the broken parts of the numeral. After selecting a pair of end points, they are joined by a line of width R_w.

After removing the dis-connectivity of numerals, we proceed for recognition. We do not compute any feature from the image. The raw images normalized into 28x28 pixel size are used for classification. The normalization procedure is discussed below.

3.1 Normalization

Normalization is one of the important pre-processing factors for character recognition. Normally, in normalization the character image is linearly mapped onto a standard plane by interpolation/extrapolation. The size and position of character is controlled such that the length and width of normalized plane are filled. By linear mapping, the character shape is not only deformed but also the aspect ratios are changed. Here we use an Aspect Ratio Adaptive Normalization (ARAN) technique [11]. For ease of classification, the length and width of a normalized image plane is fixed. In ARAN adopted by us, the image plane is not necessarily filled. Depending on the aspect ratio, the normalized image is centered in the plane with one dimension filled. Assume the standard plane is square and its side length is L. If the width and height of the input image are W_1 and H_1, respectively, then the aspect ratio (R_1) is defined by

$$R_1 = \begin{cases} W_1 / H_1 & \text{If } W_1 < H_1 \\ H_1 / W_1 & \text{otherwise} \end{cases} \qquad (1)$$

For ARAN normalization, the width and height of the normalized image, W_2 and H_2, are determined. We set max (W_2, H_2) equal to the side length L of the standard plane, while min (W_2, H_2) is determined by its aspect ratio. The aspect ratio of the normalized image is adaptable to that of the original image. Hence, the aspect ratio mapping function determines the size and shape of the normalized image. The image plane is expanded or trimmed to fit this range. The aspect ratio of the original image is calculated by equation (1). To calculate the mapping function (R_2) for the normalized image, we have used square root of the aspect ratio of the original image, given by

$$R_2 = \sqrt{R_1}.$$

To map the image f (x, y) to the new image g (x', y') we have used forward mapping to implement the normalization given by

$$x' = \alpha x \qquad y' = \beta y$$

$$\text{where} \quad \alpha = W_2 / W_1 \quad \text{and} \quad \beta = R_{2*} H_2/H_1 \quad \text{if } (W_2 > H_2),$$
$$\alpha = R_{2*} W_2/W_1 \text{ and } \beta = H_2/H_1 \qquad \text{otherwise.}$$

3.2 Neural network

Based on the above normalization, we use a Multilayer Perceptron (MLP) Neural Network based scheme [19] for the recognition of English and Bangla numerals. The present work selects a 2-layer MLP for the handwritten digit recognition. The number of neurons in input and output layers of the perceptron is set to 784 and 16, respectively. This is because the size of the normalized image is 28x28 (784), and the number of possible classes in handwritten numerals for the present case is 16. For bi-lingual (English and local language Bangla) nature of the Indian postal documents the number of numeral class is supposed to be 20 (10 for English and 10 for Bangla), but we have only 16-classes because of some common numerals in English and Bangla, and hence we have used 16 classes in the output layer of the MLP. Here 16 classes are obtained due to following. English and Bangla ‘zero’ is same and we consider these two as a single class. Also, English ‘eight’ and Bangla ‘four’ are same in the shape. Moreover, English and Bangla ‘two’ sometimes looks very similar. English ‘nine’ and Bangla ‘seven’ are also similar. Examples of some handwritten Bangla numerals are shown in Fig.9. To get an idea of some similar shaped numerals of Bangla and English, see Fig.10.

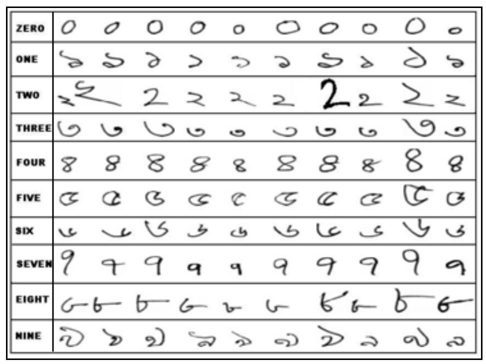


Fig. 9. Samples of Bangla handwritten numerals.

(a) **(b)**

Fig. 10. (a) English Nine and Bangla Seven, (b) English and Bangla Two.

The number of neurons in the hidden layer of the proposed network is 400, Back Propagation (BP) learning rate is set to suitable values based on trial runs. The *stopping criteria* of BP algorithm, selected for the present work, is that the sum of the squared errors for all training patterns will be less than a certain limit.

In the proposed system, we used three classifiers for the recognition. The first classifier deals with 16-class problem for simultaneous recognition of Bangla and English numerals. The second and third classifiers are for recognition of Bangla and English numerals, separately, for 10 numerals. Based on the output of the 16-class classifier, we decide the true value of pin-code. Note that Indian pin-code contains six digits. If majority of the numerals, which does not belong to common class, are recognized as Bangla by the 16-class classifier then we consider this pin-code is written in Bangla and we use Bangla classifier (the second classifier) on this pin-code to get higher recognition rate. Similarly, if the majority of the numerals are recognized as English by the 16-class classifier then we consider that pin-code is written in English and we use English classifier (the third classifier) to get higher recognition accuracy.

4. RECOGNITION OF CITY NAMES

4.1 City name recognition in a postal address reading system

As stated before, there are a reasonable number of Indian postal documents in which the pin- code is missing or just partially written. So, the name of the city or post office should be recognized to authenticate the address of the postal documents for sorting purpose. Here we consider the recognition of Indian city names written in Bangla script. Some results concerning Latin script will also be presented to get the comparative idea.

4.2 Baseline approach for word recognition

After a remarkable success of the HMMs in speech recognition [18], the model was borrowed and used with the same degree of success in handwriting recognition domain too. The power of such a model resides in its capability to track the temporal aspect of the modelled signal, which is not straightforward in a connectionist approach. While the speech can be considered as a 1D signal, the handwriting is much more complex. As the writing has its temporal aspect, the one dimensional models are less powerful to take into account this information. The HMM based models are very interesting in handwriting as are able to stock such information [8]. In the last decade, a growing interest was observed to develop new formalisms to bypass the 2D constraint.

However, direct extension of HMM into two dimensions leads to NP-hard computational complexity. Among several alternatives suggested for computationally tractable solutions, Planar HMM, Markov Random mesh and the Non-Symmetric Half Plane (NSHP) Markov chains are proposed [6]. Jeng and Woods [10] noted that NSHP chains are more appropriate than random mesh for 2-D data. Perhaps the first application of this model to handwritten word recognition on small vocabulary was due to Saon and Belaïd [22]. Essentially this model has been chosen here to recognize city names from Indian postal documents because of the complex shape on Indian words and there is no good segmentation method available on Bangla handwritten text.

Our approach of recognizing handwritten city names is based on Hidden Markov Model (HMM) and Markov Random Field (MRF). It operates on pixel level in a holistic manner over the whole word, which is viewed as outcome of the MRFs. The reason for choosing such a model is that handwriting is essentially two-dimensional in nature.

4.3 Formal description of the baseline NSHP-HMM

A complete formal description of the model can be found in [5,22]. The model works on height normalized binary image of the word analyzed from left to right and from top to bottom, considered as one possible realization of the Markov random field. For illustration, see Fig.11.

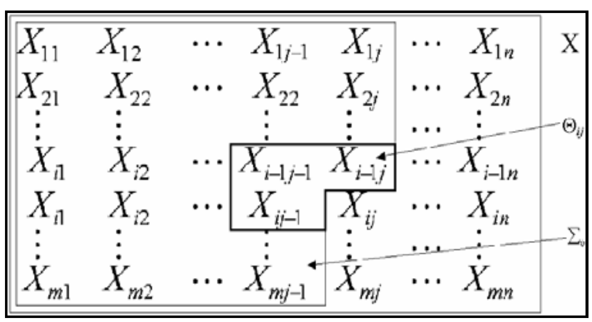


Fig. 11. The NSHP model over the lattice L.

Let the $\sum_{ij}$ be the non-symmetric half plane considering a site (i,j) defined as

$$\sum_{ij} = \{(k,l) \in L, | l < j \text{ or } (l = j, k < i)\} \qquad (2)$$

where L is the lattice of pixels defining the word image. The Markov chain is defined over a neighbourhood Θ_{ij}. For example, see Fig. 12, where various types of neighbourhood can be considered from the NSHP $\sum_{ij}$. One example of the neighbourhood Θ_{ij} of site (i,j) is given by

$$\{(i\text{-}1, j\text{-}1),\ (i, j\text{-}1),\ (i+1, j\text{-}1),\ (i\text{-}1, j)\} \qquad (3)$$

We would take a smaller neighbourhood also, as shown in Fig.12.

Now, let us define a random field $X = \{X_{ij}\}$ where $(i,j) \in L$. The column *j* of the field is denoted as X^j. Let the gray value at $X(i,j)$ be x_{ij}. Then we define conditional probability $P(X_{ij}|X_{kl})$ as the probability of realization of x_{ij} at (i,j) given that the gray value at (k,l) is x_{kl}.

The Markov process is defined to be dependent only on the neighbourhood Θ_{ij} i.e.,

$$P(X_{ij}|X_{\Sigma ij}) = P(X_{ij}|X_{\Theta ij}) \tag{4}$$

The probability of the random field X representing the gray level image denoted as $P(X)$ is written as the product from over all column fields, which in turn is written as the product of individual pixel probabilities over the column, i.e.,

$$P(X) = \prod_{j=1}^{n} P(X^j|X^{j-1}\ldots X^1) = \prod_{j=1}^{n}\prod_{i=1}^{m} P(X_{ij}|X_{\Sigma ij}) = \prod_{j=1}^{n}\prod_{i=1}^{m} P(X_{ij}|X_{\Theta ij}) \tag{5}$$

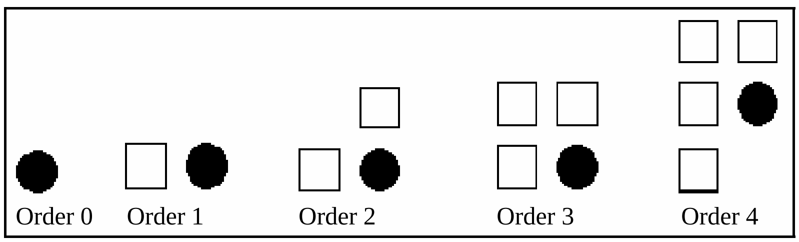


Fig. 12. Different neighborhoods for the NSHP-HMM and their order.

In order to apply the Markovian constraints, it is assumed here that the conditional probabilities are independent. A HMM denoted by λ is introduced. So, given a model λ, the probability of X is given by

$$P(X|\lambda) = \prod_{j=1}^{n}\prod_{i=1}^{m} P(X_{ij}|X_{\Sigma ij},\lambda) \tag{6}$$

The HMM is defined by the following parameters:

S={s_1,….s_N} which are N states of the model, where $q_j \in S$ denotes the state associated with column X^j. Also define the state transition probability matrix $A = \{a_{kl}; 1 \le k,l \le N\}$ where a_{kl} is the transition probability from *k* state to *l* state. The initialisation is done by the initial state probability $\pi = \{\pi_i, 1 \le i \le N\}$. Finally, there is a conditional pixel observation probability $B = \{b_{il}; 1 \le \mathrm{i} \le \mathrm{M}; 1 \le \mathrm{l} \le \mathrm{N}\}$ where

$$b_{il}(x, x_1, x_2 \ldots, x_p) = P(X_{ij} = x \mid x(\Theta_{ij}), q_j = s_l) \tag{7}$$

i.e., the probability that the current pixel is of value *x* given the neighbourhood pixels $(x_1, x_2, \ldots x_p)$ as well as the state *j* is s_l. Briefly, the model is characterized by

$$\lambda = (\Theta, A, B, \pi) \tag{8}$$

Here the number of states is initially decided from the physical problem. If *n* is the length of the analyzed word, then a value of *n/2* has been found to be a good choice for handwriting recognition. To establish the number of states, some statistics were performed on the normalized training dataset.

In other works, the number of states is fixed manually without considering shape aspect of the letters. In [9], the authors have fixed the states number to 14 considering the constant number approaches. They used the same strategy as described above, considering the total letter width estimated through the HMM for the training set, running in “forced alignment” mode. Next, the state transitions are defined. For the current problem only the strict left to right architecture is used, where transition to the same or to the forward state is permitted. Moreover, all initial transition probabilities are assumed equal i.e., $a_{ii} = a_{ii+1} = 0.5$ in order to not favour a transition or another. The image was height

normalized to have 20 rows. This value is fixed based on the trade-off between the considered information amount and the complexity of the model.
For the training phase, the goal is to determine the parameters A and B as well as π that maximize the product $\prod_{r=}^{k} P(X_r|\lambda)$ where X_r denotes a training pattern image. This is done by the well-known Baum-Welch re-estimation procedure. In this way, the model λ for each pattern class is trained sub-optimally.

Given a binary image the conditional probability of models is computed via Baye's rule from the image. This probability gives the likelihood that the image comes from that particular class. The class, for which this likelihood is maximum, is chosen as the required class. In other words, X comes from the model λ^* where

$$\lambda^* = \underset{\lambda\in\Lambda}{\operatorname{argmax}} P(\lambda|X) = \underset{\lambda\in\Lambda}{\operatorname{argmax}} \frac{P(X|\lambda)P(\lambda)}{P(X)} = \underset{\lambda\in\Lambda}{\operatorname{argmax}} P(X|\lambda)P(\lambda) \qquad (9)$$

Since $P(X)$ is constant throughout for a given image, it is dropped in the rightmost side of expression. Here Λ denotes the set of models for all classes.

The model was applied for handwritten Bangla city names extracted manually from Indian postal documents and for French bank check amounts. The image normalization was based on middle zone finding, considering the upper-line and baseline of the script.

The results obtained by the baseline system and the results presented in [5,22] have shown the limits of the NSHP-HMM model. Using just column-wise pixel information without any priori knowledge on the cutting points seems to be not sufficient to reach higher recognition performances. This insufficiency is coming from the MRF's nature. The pixel information and the neighbourhood it is carrying just a restricted amount of information, which is sensitive to the different geometrical distortion and noise. As it was shown by Choisy and Belaid in [5] the order of the neighbourhood does not influence considerably the accuracy of the system. In consequence, a higher order neighbourhood increases considerably the computational complexity of the model without a considerable score gain. Hence, some other kind of improvement should be considered like extra information implant in the model so that the complexity factor did not increase considerably.

4.4 Extension of NSHP-HMM

In order to increase the discriminating power of the model we can incorporate the information concerning the structural property. The different structural features like cutting points, ascenders, descenders, etc. can improve in different manner the system. The ascenders, descenders are well-know structural features and they can help the general word model to better distinguish between shapes as this extra information can accentuate a Viterbi path instead of another.

The challenge of this approach is how to introduce such information in the model or how to transform the model itself to accept such extra information without disturbing the baseline NSHP-HMM system.

4.4.1 Different weight mechanism

The possible structural information carried out by each pixel can be transformed into some kind of weight. This weight derived from the structural information could be descriptive for each pixel of a column (e.g. each conditional pixel probability can be weighted individually by calculating a weight for each pixel) or factorised along the column (e.g. the whole column observation probability is weighted by a value calculated in function of the different pixels' structural capacity belonging to a given column).

Let the conditional probability of the pixel (i, j) be denoted by p_{ij}:

$$p_{ij} = P(X_{ij} | X_{\Theta_{ij}}) \tag{10}$$

and the column probability:

$$P_j = \prod_{i=1}^{n} p_{ij} \tag{11}$$

Considering the equation (10) and (11), the equation (6) can be computed as follows:

$$P(X|\lambda) = \prod_{j=1}^{n} P_j \tag{12}$$

The notation used in (5-8) is similar as depicted in the Fig.13. In that case P_j denotes the column observation used by the NSHP-HMM.

Visually this weight factor can be interpreted like:

- If the weight is at pixel level, we can accentuate a pixel giving it an extra power. This can be translated in physical terms like the HMM have seen the same pixel several times or with a weight, where the weight is the importance of the pixel among the others.
- If the weight is global for the column, we can accentuate a column giving it an extra power. This can be translated in physical term that the HMM has seen the same column several times or with a weight, where the weight is the importance of the column among the others.

If such weights are applied the basic Markov constraints ($\sum_{j=1}^{N} a_{ij} = 1,\ \forall i \in S$ and $\sum_{j=1}^{M} b_j(k) = 1,\ \forall k \in V$) should be satisfied [18]. Let denote $w^{\inf}$ the general weight derived from the extracted structural feature. Different weighting mechanisms can be proposed in function of the global or local nature of the given weight.

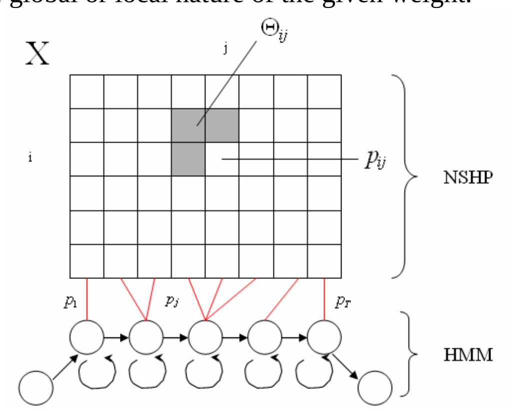


Fig. 13. A scheme of the NSHP-HMM.

Different proposals of using local weights have been made but we limit further discussion to the following equations where global weights are considered.

1. If the structural weight is global for the column j, we propose to transform the equation (11) into:

$$\overline{P_j} = \left(\prod_{i=1}^{m} p_{ij} \right) \times w_j^{\inf} \tag{13}$$

where $w_j^{\inf}$ is considered as being the weight calculated for the column j considering all the pixels (i, j) and their structural properties.

2. If we want to accentuate more the importance of the column probability j we propose to transform the equation (11) into:

$$\overline{P_j} = \left(\prod_{i=1}^{m} p_{ij}\right)^{w_j^{\inf}} \qquad (14)$$

The challenge is how to calculate this weight and how to determine the normalization factor. Generally, the weight $w^{\inf}$ can be calculated considering the quantity and the quality of the information but in our approach, we have considered just the quantitative approach without making any difference between pixels having different characteristics.

In order to obey the Markov constraints, a normalization process in necessary. As the structural information is extracted from the height normalized images, as mentioned before, the quantitative normalization is already ensured. In order to distinguish between a column where no structural information exists and a column where pixels carrying structural information, the weight $w_j^{\inf}$ for the equation (13) is calculated as follows:

$$w_j^{\inf} = \frac{1}{nbFeature + 1} \qquad (15)$$

where *nbFeature* denotes the number of pixels having a structural property (ascenders, descenders etc.) in the column j.

Considering the equation (13) the column based observation is extended for the structural-NSHP-HMM:

$$\overline{P_j} = \left(\prod_{i=1}^{m} p_{ij}\right) \times \frac{1}{nbFeature + 1} \qquad (16)$$

The $w_j^{\inf}$ in the equation (14) is calculated as follows:

$$w_j^{\inf} = \begin{cases} \eta & \text{if } nbFeature > \kappa \\ 1 & otherwise \end{cases} \qquad (17)$$

where η and κ are some parameters set to suitable values based on trial runs. In this case, the column observation described by (14) becomes:

$$\overline{P_j} = \begin{cases} \left(\prod_{i=1}^{m} p_{ij}\right)^{\eta} & nbFeature > \kappa \\ \prod_{i=1}^{m} p_{ij} & otherwise \end{cases} \qquad (18)$$

Once we have defined these new, column observations described by, equation (16) and (18), the same Baum-Welch re-estimation and Viterbi algorithm developed and described in [5,22] can be used for the structural NSHP-HMM depicted in Fig 14.

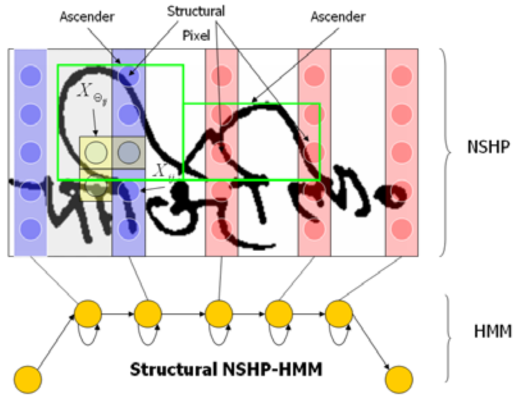


Fig 14: The structural NSHP-HMM model.

We used extra information based on structural features extracted from different word shapes, as we consider these high-level features sufficiently descriptive for handwriting. Moreover, many HWR systems use such features to discriminate different shapes. As the method is general, any other kind of information can be used instead of the information selected by us.

5. RESULTS AND DISCUSSION

5.1. Results on DAB detection

The performance of the proposed system on postal stamp/seal detection, and DAB location are as follows. We have tested our approach on 7500 postal images and noted that the accuracies for postal stamp/seal detection and DAB location are 96.16% and 98.62%, respectively. Some errors appeared due to overlapping of postal stamp/seal with the text portion of address part. Some errors also appeared due to poor quality of the images.

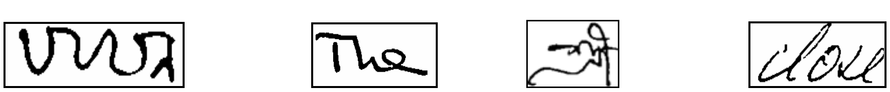
5.2. Results on script identification

For script identification experiment, we use a database of 10,342 (6100 Bangla and 4242 English) handwritten words and 1250 (700 Bangla 550 English) printed words. These data are collected from postal documents. From the experiment with handwritten data, we noticed that our proposed scheme could detect 89.78% words correctly. We have also found that the accuracy of proposed approach was 98.42% on printed text. Moreover, our handwriting identification result was 93.27% when small words (words with one or two characters) are ignored. Detailed results are shown in Table 2.

Table 2: Classification results on handwritten data.

Script	Recognized as		Rejected
	Bangla	English	
Bangla	**89.58**	7.84	2.58
English	6.17	**90.28**	3.55

The main sources of mis-recognition and rejection are small words and poor quality of postal documents. Due to presence of some small components in the upper part of the busy-zone (e.g. ) for Bangla script they sometimes are also mis-recognized as English script or rejected by the system. In addition, due to bad writing medium and poor quality of the postal documents, the words in the image are sometimes broken, resulting in mis-classification or rejection by the system. Some examples are shown in Fig. 15.



(a) **(b)** **(c)** **(d)**

Fig. 15. (a-b) Some examples of mis-classified script words (a) Bangla script classified as English. (b) English script identified as Bangla. (c-d) Some rejected scripts.

5.3. Results on pin-code box detection

We have tested our system on 7500 postal images and the accuracy for pin-code box extraction module is 97.44%. The main source of errors was due to broken pin-code box, poor quality of the images and touching of the text portion of DAB with the pin-code box.

5.4. Results on numeral recognition

For the experiment, we collected 15096 numeral data, of which 12070 data were collected from real postal documents and rest are collected from filled-in forms. Among these numerals, 8690 (4690 of Bangla and 4000 of English) were selected for training of the proposed 16-class recognition system and the remaining 6406 (3179 of Bangla and 3227 of English) numerals were used as test set. For experiment on English and Bangla individual classifier, we also collected two datasets of 10677 and 11042 numerals. We considered 5876 (6290) data for training and 4801 (4752) data for testing of English (Bangla) numeral classifiers.

The overall accuracy of the proposed 16-class classifier and individual Bangla and English classifiers on the above data set are given in Table 3. Please note that there is no rejection criterion. From the table we note that in case of Bangla classifier we obtained 2.03% better accuracy than the 16-class classifier. This is due to decrease in the number of classes and increase in the shape similarity among English and Bangla numerals. The confusion matrix of three classifiers are shown in Table 4, Table 5(a) and 5(b), respectively.

Table 3: Overall numeral recognition accuracy.

Classifier	Recognition rate for (dataset)	
	Training Set	Test Set
16-class classifier	98.31% (8690)	92.10% (6406)
Bangla classifier	98.71% (6290)	94.13% (4752)
English classifier	98.50% (5876)	93.00% (4801)

From Table 4 it can be noted that highest accuracy is obtained for Bangla numeral ‘eight’ (৮), and its accuracy is 98.54. For Bangla numeral classifier we note that highest accuracy is also obtained for numeral ‘eight’ (৮), (98.06%). For English numeral classifier we note that highest accuracy is obtained for numeral ‘zero’ (0) (97.63%). These results are shown in Table 5.

Table 4: Confusion matrix for 16-class classifier.

Numeral (data size)	Classified as															
	০	১	২	৩	৪	৫	৬	৭	৮	৯	1	3	4	5	6	7
০ (1226)	1169	2	2	14	5	8	4	5	1	1	0	0	4	8	3	0
১ (433)	1	399	4	1	2	0	5	0	0	17	1	0	0	2	0	1
২ (759)	4	8	689	1	20	6	0	3	3	2	6	3	8	1	3	2
৩ (303)	2	3	0	269	2	10	9	0	0	5	0	0	0	0	3	0
৪ (507)	3	0	2	0	476	1	0	8	0	0	4	4	1	4	3	1
৫ (246)	4	2	1	1	4	233	0	0	1	0	0	0	0	0	0	0
৬ (211)	1	2	0	9	0	3	191	0	2	1	0	0	1	0	1	0
৭ (655)	1	0	4	1	5	0	0	622	0	0	1	1	11	0	0	9
৮ (206)	1	0	0	0	0	0	0	0	203	1	1	0	0	0	0	0
৯ (206)	1	13	2	0	3	0	2	0	0	184	0	0	0	0	0	1

1 (418)	0	1	14	0	1	0	0	9	4	0	375	0	1	10	2	1
3 (226)	1	1	2	1	7	0	0	7	0	0	5	189	0	9	0	4
4 (216)	0	0	0	1	0	0	0	2	0	1	2	0	207	1	0	2
5 (289)	6	0	1	0	10	0	0	2	1	1	7	9	8	239	4	1
6 (280)	0	0	1	3	3	3	3	0	2	2	2	0	1	2	258	0
7 (225)	1	0	1	0	0	0	0	12	1	0	2	0	9	2	0	197

Here we have used only single classifier for Indian postal automation and obtained 93.0% accuracy on the English numeral collected from Indian postal documents. To get the idea of our classifier we tested our classifier on MNST data and we obtained 98.59% accuracy. Hence, we can conclude that Indian postal data quality is not as good as MNIST data.

In this present study, we also computed recognition rate of full (6-digit pin-code) pin-code level. From the experiment we found that 70.34% (75.09%) cases of a Bangla (English) pin-code all of its six-numerals are correctly recognized. In case of Bangla (English) pin-code, 25.42% (19.62%) cases one out of six-digits are mis-recognized by our system. Details result on different digit error are shown in Table 6.

Table 5: Confusion matrix of 10 class (a) Bangla and (b) English classifier.

Numeral (data size)	Classified as									
	০	১	২	৩	৪	৫	৬	৭	৮	৯
০ (1226)	1175	3	1	14	4	10	1	10	0	8
১ (433)	2	394	4	1	2	0	6	0	0	24
২ (759)	5	6	705	2	20	7	1	3	6	4
৩ (303)	3	4	0	270	1	10	10	0	0	5
৪ (507)	5	1	6	0	480	4	0	10	0	1
৫ (246)	5	2	2	0	1	232	2	0	1	1
৬ (211)	1	2	1	11	0	3	189	0	2	2
৭ (655)	1	0	6	1	5	2	0	640	0	0
৮ (206)	1	0	0	0	0	0	0	0	202	3
৯ (206)	2	11	1	0	3	0	3	0	0	186

(a)

Numeral (data size)	Classified as									
	0	1	2	3	4	5	6	7	8	9
0 (1226)	1197	0	0	2	7	7	3	0	3	7
1 (418)	0	369	16	1	3	12	3	4	0	10
2(759)	7	6	701	7	8	0	6	2	17	5
3(226)	4	5	4	192	1	4	0	5	1	5
4(216)	0	1	0	0	208	1	1	3	0	2
5(289)	7	8	3	14	6	234	4	1	11	1
6(280)	2	1	4	0	1	1	268	0	3	0
7(225)	0	3	1	0	9	1	0	200	0	11
8(507)	4	4	5	4	2	4	3	1	475	5

(b)

9(655)	2	3	4	1	14	0	1	7	2	621

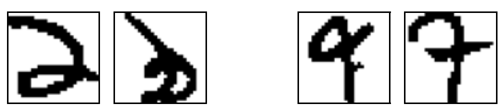
Table 6: Full pin-code recognition result.

Error in number of numeral	Recognition rate at pin-code level	
	Bangla script	English script
0	70.34%	75.09%
1	25.42%	19.62%
2	3.59%	4.78%
3	0.85%	0.50%
4	0.00%	0.00%
5	0.00%	0.00%
6	0.00%	0.00%

From the experiment, we noted that the most confusing numeral pair was Bangla 'one' and Bangla 'nine' (shown in Fig. 16(a)). They do confuse in about 6.3% cases. Their similar shapes rank the confusion rate at the top position. The second confusion pair is Bangla seven and English seven (see Fig. 16(b)) with confusing rate of 5.3%. We did not incorporate any rejection scheme in the proposed system, which we plan to add in the future.

To have a comparative idea about the other published work, we compared our results with a recently published work on Bangla postal numeral recognition due to Wen et al. [26]. They received 95.05% on Bangla numeral combining different classifiers, whereas we have obtained 94.13% using a single classifier.

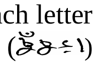
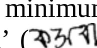
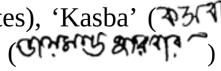
To get an idea about processing time, we computed the total time required from binarization to recognition. Our system with Pentium-IV machine with 2.4 Ghz Processor and 256 Mb RAM needs 2.15 seconds (average) for processing an Indian postal document.



(a) **(b)**

Fig. 16. Examples of some confused handwritten numeral pairs. (a) Bangla one and nine (b) Bangla seven and English seven.

5.5. Results on word recognition using the baseline NSHP-HMM

For the experiment of word recognition, we have considered Indian city names written in Bangla script. Here a total of 76 city name each at least of 100 samples with a total of 8463 city names have used for the experiment. In order to optimise the NSHP-HMMs model for each word class, we have generated statistics concerning the letter length in the different words in order to estimate the average length of each letter in the data set. The minimum numbers of NSHP-HMM states are in the model of 'Chuchura' () (18 states), 'Kasba' () (19 states) and the maximal number of states is in the model of 'Dimonharber' () (54 states).

The normalization of the words is performed just in height since left-right NSHP-HMM model can take care of the width normalization, as reported in [5]. The differential normalization used is based on finding the middle zone of the script. This zone is estimated throughout the product of horizontal projection and the horizontal run-lengths of the image. Once the middle zone is established, the upper zone, the lower zone and the middle zone are mapped in the same manner. A threshold value has been established empirically in case when the middle zone height is not sufficient. In that case, the given image is discarded from the data corpus. For other normalization approaches see [17]. In Fig. 17 we present some Bangla images and their normalized form.

Original Image	**Normalized image**

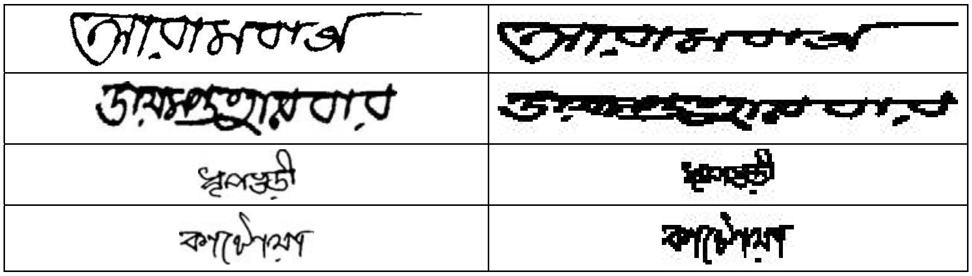


Fig. 17. Results of height normalization.

It was necessary to discard the images, since just in 2% of the cases the middle zone's height was not sufficient. In such cases, the word was written really in a slant manner.

For the NSHP-HMM observations, different pixel neighbourhoods can be used. We have considered the 3rd order neighbourhood since the results reported by Choisy and Belaïd [5] showed that this architecture maximally preserves the graphical information in the pixel columns and reduces the complexity of the system used by Saon and Belaïd [22]. For a model having *N* states, analysing *Y* pixels in a column, the memory complexity of the system is $O([N+Y*2^n])$ where *n* denotes the order of the neighbourhood. As we can see, the complexity will grow exponentially in function of the neighbourhood order and in linear order for the parameters like *N* and *Y*.

The training of the system is based on the well-known Baum-Welch optimization procedure. During the training, a good stabilization was observed which indicate the convergence of the system.

The overall recognition accuracy for different vocabulary size is given in the Table 7. The results show that the accuracy of the system decreases when size of the classes increases. The 86.44% recognition accuracy is achieved when 76 classes are considered. This result is comparable with the state of art results reported for such a vocabulary size using holistic, segmentation-free approaches. In order to get an idea about the recognition accuracy for each word class, we report in the Table 8 the accuracy achieved for each word model.

Table 7: Overall recognition accuracy on the training and test set of Bangla data for different vocabulary size.

Vocabulary size	**Recognition rate**	
	Training set	**Test set**
30	93.49%	92.04%
40	93.41%	90.38%
50	94.00%	88.97%
60	94.02%	88.27%
70	94.58%	87.30%
76	94.83%	86.44%

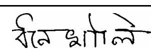
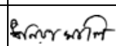
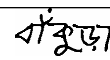
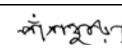
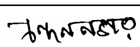
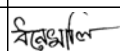
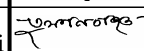
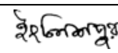
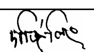
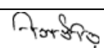
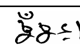
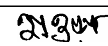
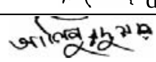
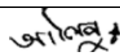
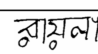
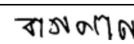
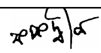
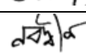
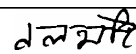
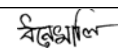
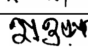
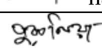
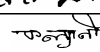
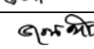
Table 8: Recognition accuracy of individual city name.

Word name	**Ac. (%)**	**Word name**	**Ac. (%)**	**Word name**	**Ac. (%)**
dhanekhali	82.35	chandannagar	47.06	bagnan	91.18
srirampore	91.18	bankura	84.85	tarokeswar	91.18
bishnupur	93.94	uluberia	100	rayganj	97.06
dhaniakhali	94.12	bardhaman	87.88	gangarampur	97.06
raina	88.24	islampur	91.18	kalna	67.65
karandighi	97.06	durgapur	94.12	patrasayar	94.12
seuri	87.10	asansole	100	kantoa	91.18
rampurhat	100	memari	93.75	chittaranjan	96.97
bolepur	88.24	santiniketan	97.06	nalhati	87.88
murshidabad	96.88	beldanga	82.35	rajnagar	82.35

basirhat	94.12	barasat	97.06	kasba	88.24
jalangi	73.53	jalongi	85.29	sodepore	94.12
panskura	76.47	jangipore	73.53	farakka	73.53
tomlook	79.41	bongao	79.41	malda	82.35
englishpore	94.12	harischandrapore	91.18	kanshipore	91.18
dimonharber	88.24	purulia	85.29	manbazer	90.62
namkhana	85.29	raghunathpore	88.24	sonarpore	85.29
darjiling	85.29	kalimpong	67.65	alipurduwar	85.29
alipur	94.12	ranaghat	85.29	coachbihar	91.18
chakda	85.29	shantipur	94.12	bali	93.94
mathabhanga	67.65	nabadwip	94.12	kalighat	85.29
kakdwip	73.53	arambag	97.06	jhargram	70.59
kanthi	85.29	barrackpore	91.18	jalpaiguri	91.18
karsiang	85.29	dhupguri	87.88	nakshalbari	91.18
tuphangange	67.65	kalyani	84.85	chuchura	70.00
howrah	61.76				

From the experiment performed we have noted that the main confusion occurs in cases where the word shape is almost similar ((dhanekhali, dhaniekhali), (jalangi, jalongi)) or in cases where a considerable part of the word shape is similar (like in kakdwip, nabadwip). See Table 9 for more details on the confusion. The other confusion can be explained with the great variability of the letters and inter-letter connections, because Bangla has 350 different letters and shape modifiers instead of the 52 letters used in Latin scripts. In Table 9, for each word class its most often confusing class is shown.

Table 9: Example of some confusing city-name pair obtained from the word recognizer.

Original Image	**Maximally mis-recognized as**	**Original Image**	**Maximally mis-recognized as**
 dhanekhali	 dhaniakhali	 bankura	 panskura
 chandannagar	 dhanekhali	 tuphangange	 englishpore
 darjiling	 seuri	 chuchura	 howrah
 alipurduwar	 alipur	 raina	 bagnan
 kakdwip	 nabadwip	 nalhati	 dhanekhali
 howrah	 purulia	 kalyani	 jalangi

5.6 Results on word recognition using the structural NSHP-HMM

In order to test the system capabilities, two datasets were used. The first one is the Bangla city names described earlier and the second dataset is Latin one, so called Service de Recherche Technique de La Poste (SRTP) dataset [22] containing handwritten French bank check amounts. In SRTP, dataset 7031 images are distributed non-uniformly in 26 classes. The 26 classes correspond to the different French words describing the different legal amounts. Similarly, as in the case of the Bangla city names, 66% of images served to train the system and the remaining 34% were used for test purpose. Using the similar system as for the Bangla city names recognition, the achieved accuracy for the SRTP dataset is 85.95%. The results obtained by the different improved (extended) observations on similar training and test data are summarized in Table 10.

Table 10: Results using the structural information.

Method	**SRTP (26 class)**	**Bangla (76 class)**
Method based on Eq.16	87.52%	86.80%
Method based on Eq.18	86.39%	86.52%

The achieved accuracy for the SRTP dataset using the column observation given by the equation (1) is 87.52%, which is an important gain in comparison with the baseline system's results. Meanwhile the same observation for the Bangla dataset gives just 0.36% of improvement. The difference is due to the nature of the scripts and the used structural features. While in case of the SRTP bank cheque dataset, the words are Latin words so the notion of ascender/descender is clearly distinguishable; the same notion has not the same signification in the case of the Bangla script. In order to reach higher results for Bangla script, some other kind of structural features (like water reservoir features, matra feature) should be extracted.

Generally, to get more important improvements, more adequate structural information should be extracted, like convex and concave sectors, cross points, cutting points, etc which better describes the given script. Extracting a huge variety of features, the normalization process can be also refined as different weights can be assigned to the different features in terms of their discriminating power.

6. CONCLUSION

A system towards Indian postal automation is discussed here. In the proposed system, at first, using RLSA, we decompose the image into blocks. Based on the black pixel density and number of components inside a block, non-text block (postal stamp, postal seal etc.) are detected. Using positional information, the DAB is identified from the text block. Then the line and word from the DAB are segmented to recognize the script in which the words are written. Next, pin-code box from the DAB is detected and numerals from the pin-code box are extracted. Pin-code digits are recognized for postal sorting according to the pin-code of the documents.

For the words written in Bangla script, we recognize them to verify the result obtained by pin-code recognition module. After a differential height normalization of word images, a model of discriminative stochastic approach has been used to recognize them. A general improvement mechanism based on weighting derived from the structural properties of the different column has been showed.

Please note that because of poor quality of Indian postal documents, writing device/medium, wide range of educational background and multi-script culture, image qualities of Indian postal dataset is not good. Because of this reason, we did not get high accuracy. At present the whole system has 51.41% accuracy on postal documents. Moreover, Bangla text contains many complex shaped compound characters and this also degrades the recognition accuracy.

Acknowledgement: Partial financial support by Indo-French Centre for the Promotion of Advanced Research (IFCPAR) is greatly acknowledged. We are also thankful to the Indian Postal department for providing us space and facility to scan images of real postal documents. One of the authors (B. B. Chaudhuri) would like to thank Jawaharlal Nehru Foundation for partial support in the form of fellowship.

REFERENCES

[1] S. Basu, C. Chaudhuri, M. Kundu, M. Nasipuri, and D. K. Basu, "A two-pass approach to pattern classification", In Proc. of 11th ICONIP, pp. 781-786, 2004.

[2] U. Bhattacharya, T. K. Das, A. Dutta, S. K. Parui and B. B. Chaudhuri, "A hybrid scheme for handwritten numeral recognition based on Self Organizing Network and MLP ", International Journal of Pattern Recognition and Artificial Intelligence, Vol. 16, pp. 845-864, 2002.

[3] J. Cai and Z. Q. Liu, "Integration of structural and statistical information for unconstrained handwritten numeral recognition", IEEE Trans. on PAMI, Vol. 21, pp. 263-270, 1999.

[4] B. B. Chaudhuri and U. Pal, "A complete printed Bangla OCR system", Pattern Recognition, Vol. 31, pp. 531-549, 1998.

[5] C. Choisy and A. Belaïd, “Handwriting recognition using local methods for normalization and global methods for recognition”, In Proc. of 6th ICDAR, pp. 23-27, 2001.
[6] C. Choisy and A. Belaïd, “Cross-learning in analytic word recognition without segmentation”, International Journal on Document Analysis and Recognition”, Vol. 4, pp. 281-289, 2002.
[7] A. K. Dutta and S. Chaudhuri, “Bengali alpha-numeric numeral recognition using curvature features”, Pattern Recognition, Vol. 26, pp. 1757-1770, 1993.
[8] A. El-Yacoubi, M. Gilloux, R. Sabourin, and C. Y. Suen, “An HMM-based approach for off-line unconstrained handwritten word modelling and recognition”, IEEE Trans. on PAMI, Vol. 21, pp. 752-760, 1999.
[9] S. Gunter and H. Bunke, “Optimizing the number of states, training iterations and Gaussians in an HMM-based handwritten word recognizer”, In Proc. of 7th ICDAR, pp. 472-476, 2003.
[10] F. C. Jeng and J. W. Woods, “On the relationship of the Markov mesh to the NSHP Markov chain”, Pattern Recognition Letters, Vol. 5, pp. 273-279, 1987.
[11] C. L. Liu, K. Nakashima, H Sako, H. Fujisawa, “Handwritten digit recognition: investigation of normalization and feature extraction techniques”, Pattern Recognition, Vol. 37, pp. 265-278, 2004.
[12] U. Pal and B. B. Chaudhuri, “Automatic recognition of unconstrained off-line Bangla hand-written numerals”, In Proc. of LNCS-1948, pp 371-378, 2000.
[13] U. Pal and B. B. Chaudhuri, “Indian script character recognition: A survey”, Pattern Recognition, Vol. 37, pp.1887-1899, 2004
[14] U. Pal and S. Datta, “Segmentation of Bangla unconstrained handwritten text”, In Proc. of 7th ICDAR, pp. 1128-1132, 2003.
[15] U. Pal and P. P. Roy, “Multi-oriented and curved text lines extraction from Indian documents”, IEEE Trans. on Systems Man and Cybernetics - part B: Cybernetics, Vol. 34, pp. 1676-1784, 2004.
[16] P. Palumbo, P. Swaminathan, and S. Palumbo, “Document image binarization: Evaluation of algorithms”, SPIE Applications of Digital Image Processing IX, Vol. 697, pp. 278-285, 1986.
[17] R. Plamondon and S. N. Srihari, “On-line and off-line handwritten recognition: A comprehensive survey”, IEEE Trans. on PAMI, Vol. 22, pp. 62-84, 2000.
[18] L. R. Rabiner, “A tutorial on Hidden Markov Models and selected applications in speech recognition, In Proc. of IEEE, Vol. 77, pp. 257-286, 1989.
[19] K. Roy, C. Chaudhuri, P. Rakshit, M. Kundu, M. Nasipuri and D. K. Basu, “A comparative assessment of performances of the RBF Network and the MLP for handwritten digit recognition”, In Proc. of CODIS, pp. 540-543, 2004.
[20] K. Roy, U. Pal and B. B. Chaudhuri, “A system for joining and recognition of broken Bangla numerals for Indian postal automation”, In Proc. of 4th Indian Conference on Computer Vision, Graphics and Image Processing, pp. 641-646, 2004.
[21] K. Roy, S. Vajda, U. Pal, and B. B. Chaudhuri, “A system towards Indian postal automation”, In Proc. of 9th IWFHR, pp. 580-585, 2004.
[22] G. Saon and A. Belaïd, “High performance unconstrained word recognition system combining HMMs and Markov Random Fields”, International Journal of Pattern Recognition and Artificial Intelligence, Vol. 11, No. 5, pp. 771-788, 1997.
[23] Z. Shi and V. Govindaraju, “Skew detection for complex document images using Fuzzy runlength”, In Proc. of 7th ICDAR, pp.715-719, 2003.
[24] F. M. Wahl, K. Y. Wong, R. G. Casey, “Block segmentation and text extraction in mixed text /image documents”, Computer Graphics and Image Processing, Vol. 20, pp. 375 - 390, 1982.
[25] T. Wakabayashi, S. Tsuruoka, F. Kimura, and Y. Miyake, “Increasing the feature size in handwritten numeral recognition to improve accuracy”, Systems and Computers in Japan, Vol. 26, pp 35-43, 1995.
[26] Y. Wen, Y. Lu and P. Shi, Handwritten Bangla numeral recognition system and its application to postal automation, Pattern Recognition, Vol. 40, pp. 99-107, 2007.
[27] P. Wunsch and A. F. Laine, “Wavelet descriptors for multi-resolution recognition of hand-printed digits”, Pattern Recognition, Vol. 28, pp 1237-1249, 1995.
[28] D. Yu and W. Lai, “Analysis and recognition of broken handwritten digits based on morphological structure and skeleton”, International Journal of Pattern Recognition and Artificial Intelligence Vol. 19, pp. 271-296, 2005.
[29] L. Zhou, Y. Lu and C. L. Tan, Bangla/English script identification based on analysis of connected component profiles, In Proc. of LNCS-3872, pp. 243-254, 2006.